International Workshop IFIP WG 3.2
on
Informatics Curricula for the 1990's
(April 1990)

- Short Paper -

Architectural Guidelines for the Curricula:

3 Layers
3 New Dimensions
2 Basic Orientations
Different Levels in the Topics

Author: Fernando Sáez Vacas
Depto. Ingeniería de Sistemas Telemáticos
Escuela Técnica Superior de Ingenieros de Telecomunicación
Universidad Politécnica de Madrid
Ciudad Universitaria, s/n. 28040 Madrid
e-mail: fsaez@dit.upm.es
fax: +34 1 2432077

1. Introduction

Today, designing informatics curricula is a major problem. As a technology, informatics is experiencing a dramatic evolution, with its rapidly expanding areas of application and ever-increasing impact on society. At first glance, it seems that if we might need several different curricula for facing very different practical educational situations, and not just one or two curricula.

The current deep discrepancies among some of our more prestigious computer scientists in relation to the focus of informatics education are in fact a form of recognition of this necessity and at the same time, proof we are at a turning point.

Everyone today accepts that prior to designing and constructing a family of computers one must design its architecture. Similarly, I believe that before entering into a discussion about objectives and contents of computing science curricula, we first need to design the bases for a variety of curricula.

This paper presents some general ideas to be discussed the main points of which are: (1) to underline the complexity of the problem undertaken, proposing a **curriculum architecture composed of layers, vectors or dimensions, orientations and levels**; and (2) to provide a framework for guiding the construction of the aforementioned variety seen as a combination of the pertinent values of these four factors.

2. Three nested layers

One method for coping with the complexity of a phenomenon or process is to model it by means of conceptual distinctions or variables which capture its basic traits. The phenomenon here under study is **the way to design a specific curriculum for each of many different situations**, and our distinctions will be layers, dimensions, vectors and levels.

We propose that any curriculum must offer with greater or lesser intensity three layers of topics, whose general contents are displayed below:

- the **essential** layer
- the **instrumental** layer
- the **experiential** layer

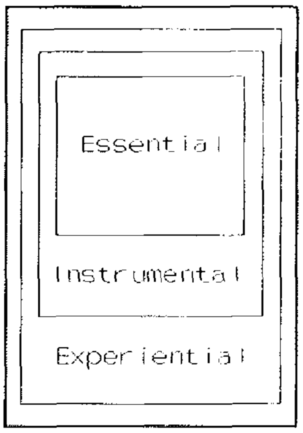

Figure 1. Curricula layers.

The **essential layer** copes with fundamental topics related to information and its processing: what is information; how is it represented, how is it computed, how is it stored, how is it transmitted, how is it measured, how is it organised as data sets; what are the limits of computations; what are the basic mechanisms to do all the previously mentioned operations at machine levels, at symbolic computing level, at data structure levels, etc.

The **instrumental layer** is the realm of the physical, logical and technological instruments with which we can integrate systems to solve a lot of information problems: languages, compilers, processors, circuits, devices, operating systems, data transmission protocols, data bases management systems, coders, computer and network architectures, etc.

The **experiential layer** represents the application areas where those instruments become purposeful. Some examples are the fields of Artificial Intelligence, Management Information Systems, Process Control, Image Processing, Computer Graphics, C.A.D., C.A.M., and many other more or less computerised fields.

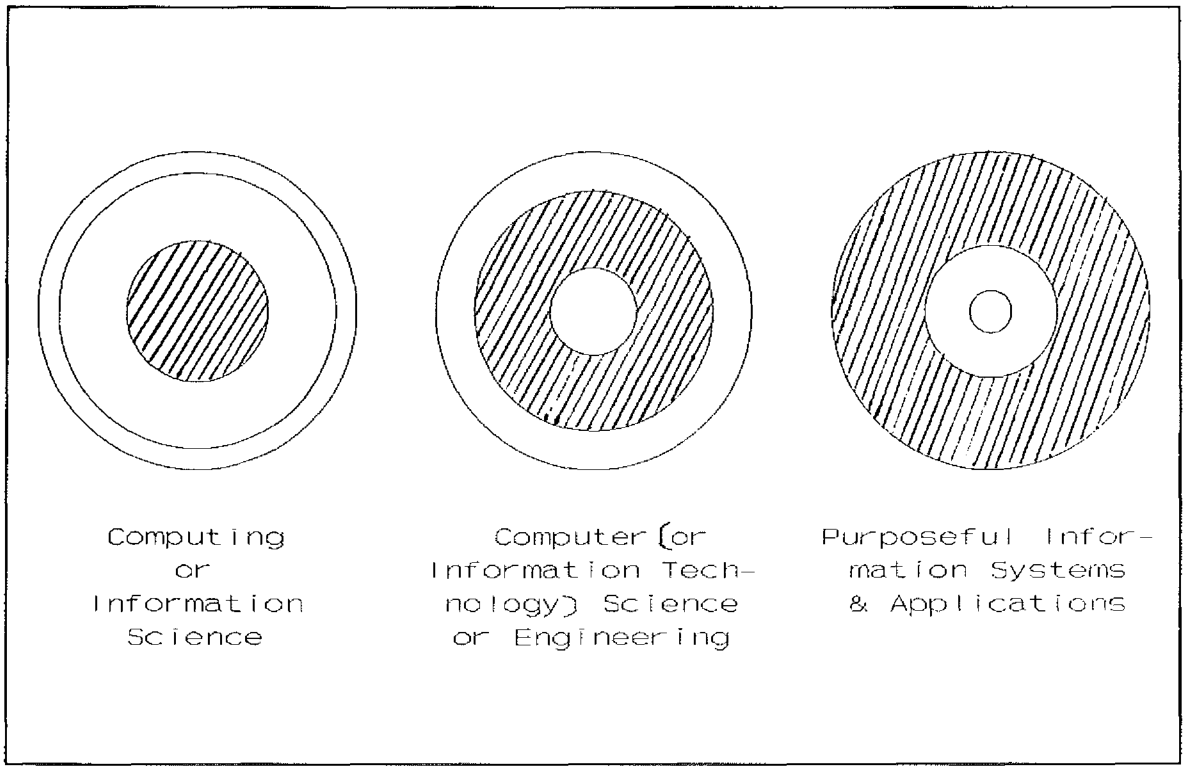
Computing or Information Science
Computer (or Information Technology) Science or Engineering
Purposeful Information Systems & Applications

Figure 2. Three archetypical curricula.

It is theoretically possible to conceive of many curricula structured around these three layers. **Each situation would require a peculiar density and composition configuration of the layers.** We can graphically represent this approach by a concentric circles model whose ring thickness tries to stress the idea of relative emphasis (perhaps quantified in number of students credits) accorded to each layer.

Figure 2 illustrates as an abstract example three extreme curricula profiles.

Let us take a concrete example: Software Engineering. S.E. is a very special case, and one which is currently controversial. Its broad scope triggers the possibility of varied curricula profiles falling somewhere within the scope of the second and the third ones represented in figure 2, depending first on the nature of the software engineered (belonging to the instrumental or to the experiential layer) and secondly on the importance accorded to the organisational areas (Software Engineering Project Management).

Dijkstra, Wirth, Gries, Parnas and other distinguished authors, among whom are found outstanding contributors to the advancement of the science and art of programming, are very critical of the state of the art of S.E. While Software Engineering is oriented toward anindustrial type of programming, these authors defend the essentials of programming. Briefly, they are proposing a part -and only a part- of a Computing (or Information) Science curriculum profile (left graphic in figure 2).

In conclusion, Software Engineering and Computing Science are two different things; but because they are very closely related there is evidence that a S.E. curriculum must be constructed with its essential layer strongly inspired in the recommendations of Computing (in this case, Programming) Science.

3. A curriculum is composed of several curriculum vectors

Now we have the problem of filling u'ı the layers. I propose to do it articulating their contents around certain lines which we can imagine being symbolised by circle radii or vectors in the model above.

If you imagine this model as a Kiviat graph you can easily understand that the vectors will have different values (included zero intensity) depending on the concrete curriculum. But more important is that each curriculum vector must be viewed as a structure of topics and credits. That is the case of the topical areas considered in the ACM Curriculum'78 Recommendations or in the coming Undergraduate Curricula in Computer Science and Engineering in preparation by the ACM and the IEEE Computer Society. A concrete curriculum will be the result of a concrete composition of all pertinent curriculum vectors.

It seems to me that to clearly define the principal curricula vectors and to elaborate a large and hierarchised curriculum of each of these vectors is an urgent task for the scientific computer community.

An example of a curriculum vector would be mathematics. As an abstract modeling language, mathematics is one of the most powerful tools for managing the complexity of computing constructs, and by its relation with the very nature of the essential layer becomes perhaps the most long range and influential curricula vector.

4. Three new dimensions

In my opinion there are three new emergent vectors that must be included in every curriculum in the convenient proportions. They are:

a) **Integration** of all types of **information processing**, particularly computing and communication.
b) **Complexity**, a constant and growing trait in the designing, construction, application and consequences of this technology.
c) **Human** impact and interfacing.

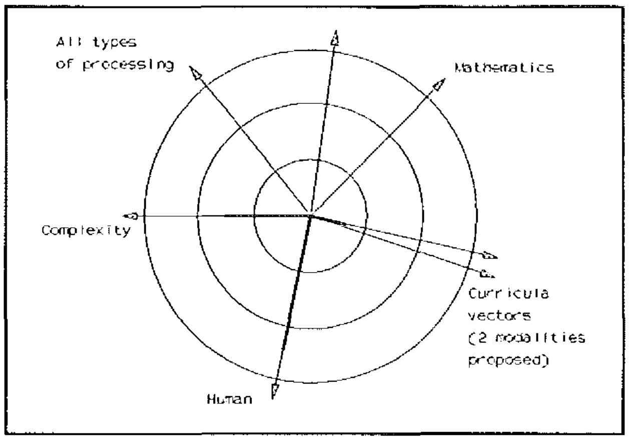

Figure 3. Curricula layers and vectors.

Let us examine briefly some of the reasons for my proposal.

Concerning the first point, I believe it is no longer realistic to study only computing. The pervasive convergence of all classes of information representation and processing must also be wholistically implemented in the curricula.

Another fact difficult to be denied is the **quasi-exponential increase of complexity**. This aspect is unanimously stressed by authors and practitioners: the complexity of a program; the complexity of an integrated circuit like a powerful microprocessor; the complexity of highly parallel computers; the complexity of managing a software development project; the computational complexity; the complexity of implementing an information system; the complex interactions between technology and economy; etc.

In each of those situations a part of the complexity is in the observer mind, so we can conclude that an inadequate and overly specialised education of the observers is a complexity generator.

These observers, that is the informatics scientists and professionals need to be educated in the techniques and abstract methods of complexity management: hierarchies, abstractions, systems thinking, problem solving, modeling, formal mathematics, uncertainty, etc.

Finally the **profound impact of this technology on man and society** overflows the fields of psychology, sociology, economy or philosophy to increasingly embody the profiles of our knowledge, experience and responsibility. To illustrate this phenomenon we can cite subjects such as computer human interfacing, conviviality of technological tools, ethics, systems vulnerability, cognitive approaches, information polution, etc.

5. Two basic orientations

Above I established that curriculum vectors are themselves curricula obeying the classical approach of structures of topics, but now I specify that they must be organised by hierarchied levels in a way that might make it possible to select the relative importance of the topics according to the orientation of each concrete curriculum.

Obviously each of the many possible experiential fields would require a special topics choice, so, if it were viable, topics could be organised in the following manner: their contents specified by modules, the modules being designed with reference to some sort of levels of detail -i.e. introductory, intermediate and advanced-, to the layers and to some other attributes (see figure 4). Is the ACM and IEEE Computer Society Joint Task Force doing something alike?.

But all this structuring would be very complicated to begin with, so I suggest concentrating our efforts on two points: (1) the definition of the curriculum vectors only for the essential and instrumental layers; (2) the selection of two basic orientations for designing the vectors.

When I propose two principal orientations I am trying to simplify making a sharp distinction with respect to the addressees of curriculum, which may stand for technology **producers** or for technology **users**. For the sake of clarity, the producers -designers, developers, engineers- would be served by the first and second of the extreme situations depicted in figure 2, and the users by the third.

The main characteristic of a producer is his orientation to the tasks of inventing, designing or constructing informatics instruments or theories. In that sense he or she needs to learn and to manage both descriptive and constructive languages in the most general sense. Users apply those basic instruments, so generally they need to know them only from a functional and descriptive perspective.

Let us take a look at an example: the operating systems. It is well known that they can be studied from a design point of view or from a usage point of view, and both approaches are radically different.

6. Conclusions

Facing the variety of curricula needs in the field of informatics it is indispensable to respond with an **architecture capable of generating a variety of curricula**. In this paper I have proposed the guidelines of such an architecture which are synthesized in figure 4.

Moreover the convenience of incorporating in the curricula three new dimensions was suggested: (a) integration of all types of information processing, and not only computing, (b) complexity, as an abstract area in itself, and (c) human impact and interfacing.

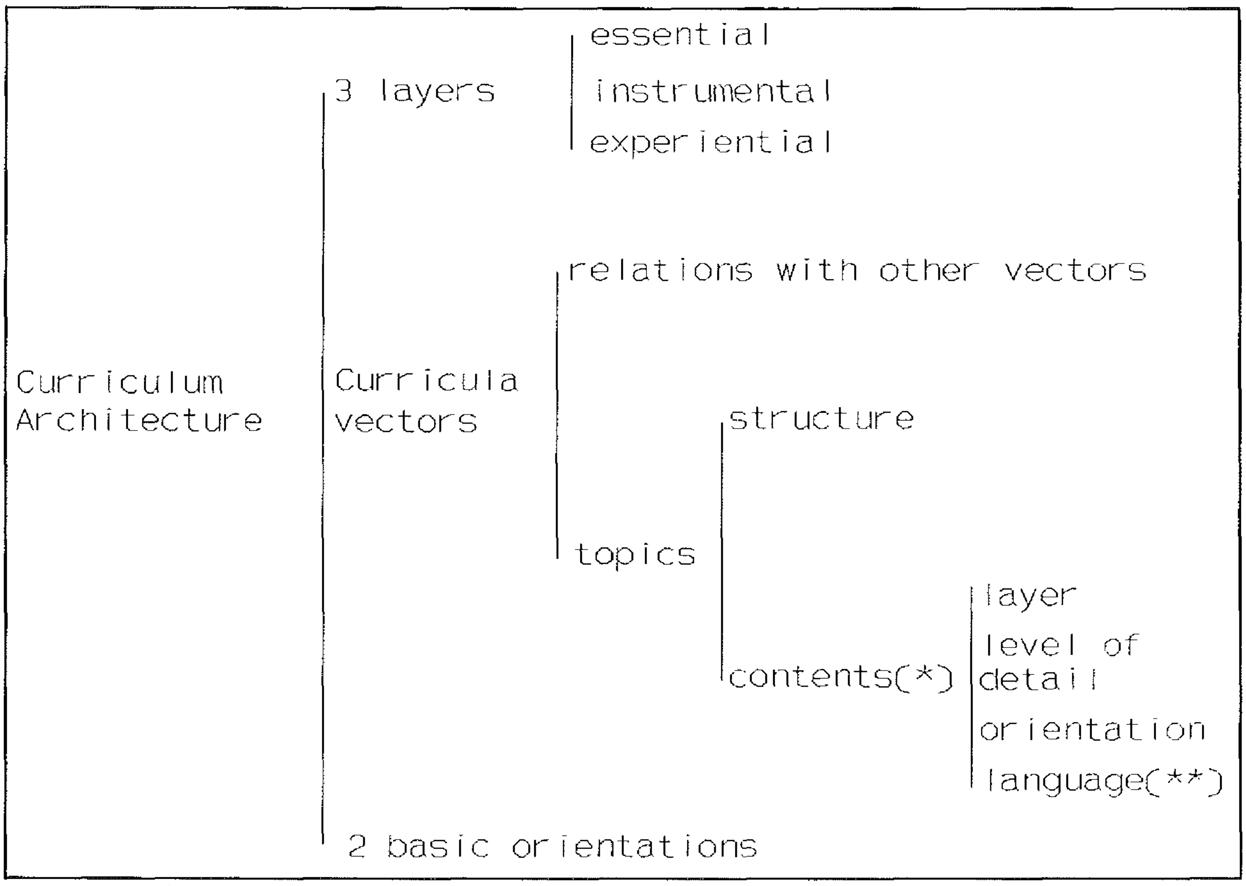

Figure 4. Curriculum architecture. (*) The hierarchy of levels will be governed by the parameters shown to the right. (**) By "language" I mean a kind of representation or formalisation.

The scientific community has much work ahead to define the curricula for the 90s, but the debates and discrepancies already published also show that, once begun, the process will not be easy.